

Tehtävä 1

Oletetaan, että pelin suorittamille toimenpiteille (tekoälyn ajaminen, pelimaailman päivitys, jne.) on määritelty yhteinen yliluokka `Game_Task`. Luokassa on määritelty puhdas virtuaalifunktio `perform()`, joka suorittaa ko. toimenpiteen:

```
virtual void Game_Task::perform() = 0;
```

Kaikki toimenpiteet ovat toisistaan riippumattomia (eivät käytä samaa dataa tms.).

Peli ylläpitää vektoria toimenpiteistä. Toimenpiteet suoritetaan peräkkäin silmukassa:

```
std::vector<Game_Task*> tasks;
...
int number_of_tasks = tasks.size();
for (int i = 0; i < number_of_tasks; i++)
{
    task[i]->perform();
}
// Continue only after all tasks are complete!
```

Hahmottele, miten em. silmukka voitaisiin rinnakkaistaa käyttäen

- a) `std::thread` -luokkaa (luo esim. kaksi säiettä, jotka käyvät toimenpide-vektorin läpi puoliksi)
- b) asynkronisia funktiokutsuja (kutsujen määrää ei tarvitse rajoittaa)

Jos/kun konkreettisesti koodaat ratkaisun, niin luo yliluokka `Game_Task` ja sille muutama aliluokka, joiden `perform`-funktio kuluttaa sopivasti prosessori-aikaa johonkin. Mittaa rinnakkaistamisesta saamasi suoritusaikahyöty.